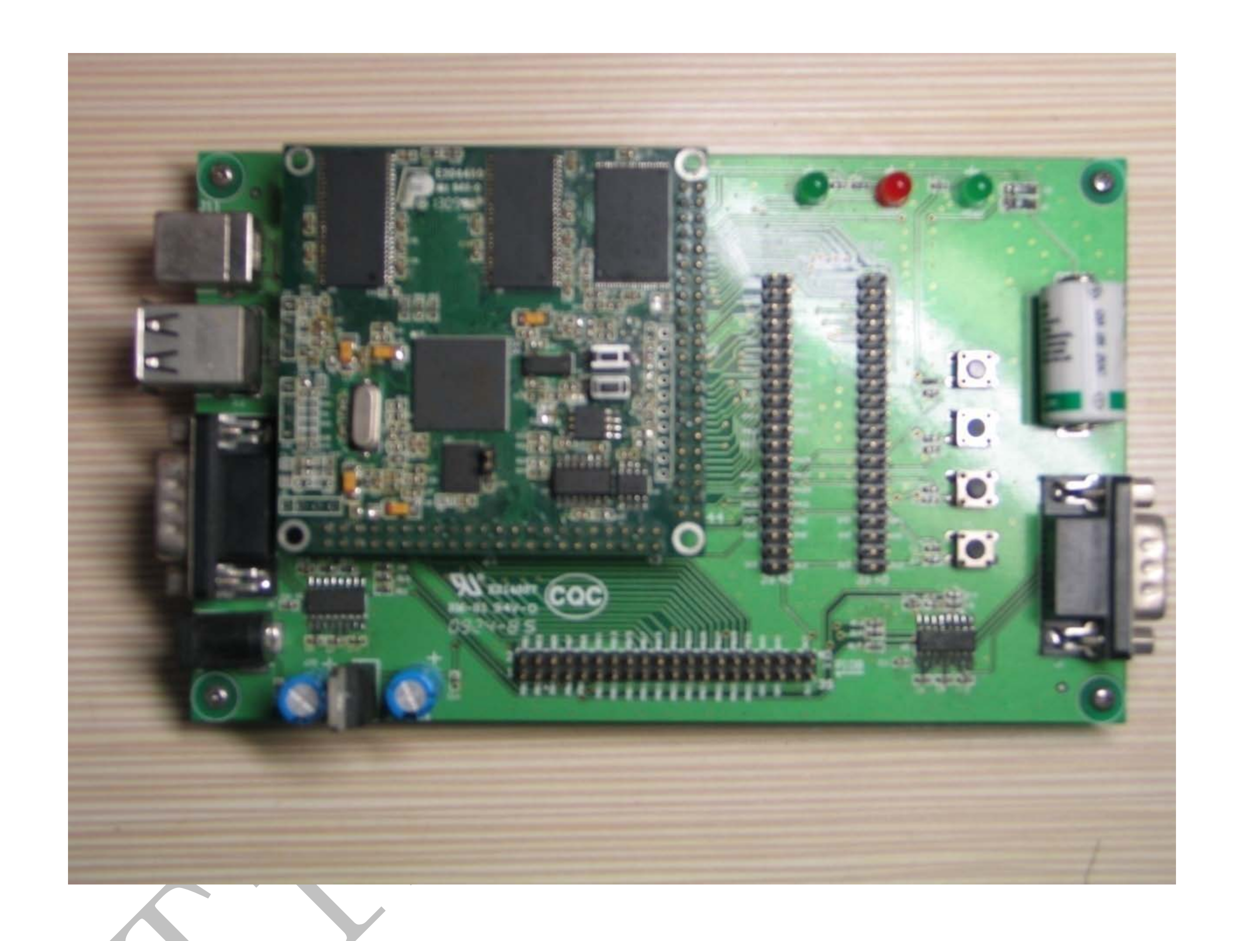
Tiny-linux使用说明书

（Ver1.02）



天鹰科技期待与您的合作
支持博客：http://tony3247.cublog.cn
支持QQ： 645076297
支持邮箱：**tianying2168@163.com**

团购价：400元
零售价：450元

首先非常感谢您使用**Tiny-linux**系统，您获得的不仅仅是一套开发板，而是一套高效的**linux**开发、调试、代码管理的方案，希望我们双方有更深层次的合作。

在使用我们的开发板之前，请阅读下面的注意事项。

- 开发板的左下角是电源部分，由于开发板的功率比较大，该部分会有可能发烫，使用者请注意。
- 使用者请认真阅读完说明书之后再对开发板进行操作，避免对开发板造成物理损害。
- 开发板支持无线和有线网卡两种通信模式。
- 开发板提供详细的技术支持，技术支持QQ：645076297，技术支持邮箱：tianying2168@163.com

开发板由两个部分组成，即核心板与扩展板，主要的功能芯片都在核心板上，核心板为6层板，这样可以有效地减少干扰。开发板支持的资源如下：

- CPU：AT91SAM9260 BGA VERSION B
- SDRAM: ISSI IS42S16100G 2片，64M
- NAND: NAND01GW3B2AN6
- DATAFLASH：ATMEL AT45DB642D
- EEPROM: ATMEL 24A512
- RTC:EPSON RTC8025
- 温度检测：TCN75

开发板采用最新的linux2.6.24内核，在官方内核版本上添加相应的芯片驱动，为提高系统的启动速度，开发板一改往常的BOOTSTRAP+UBOOT+LINUX+FILESYS的格式，使用bootstrap直接启动linux内核，可以将linux的启动时间缩短到7秒左右，如果您需要额外的技术支持，可以联系我们，开发板软件资源包括:

- Bootstrap，16位总线宽度
- UBOOT 1.3.4
- LINUX 2.6.24
- BUSYBOX 1.13.2
- FILE SYSTEM Tiny-System
- 有线网卡驱动
- 无线网卡驱动
- MTD-Utils工具。

编译环境资源，我们提供了一套完整的linux研发管理的软件，主要包括：

- 交叉编译器：arm-2007q1.tar为支持最新的arm9指令集, gcc-3.4.5-glibc-2.3.6.tar为稳定的广泛使用的版本
- Sam-ba:我们提供两个版本，windows版本和linux版本，我们在官方的基础之上进行了更改，使其支持16bits的数据总线。并支持yaffs与jffs等文件系统的一键烧写。
- Eclipse:该软件是我们推崇的linux下C/C++最优秀的集成开发环境，他强大的功能和集成的调试环境会让你觉得就像在做windows的MFC编程一样。并我们提供JAVA虚拟机，能够在目标板上完成JAVA编程。

- Smartsvn:该软件是我们推荐使用的版本管理工具，它能对多人团队开发的软件很好的进行软件版本管理

Tiny-Linux主要由两个部分组成，6层核心板和2层扩展板，下面对两者上比较重要的部件进行介绍。

核心板：

- J12：DATAFLASH的cs引脚短接跳线帽，短接表示支持dataflash。
- BP1：系统手动复位键，方便用户的调试。
- BP2：系统唤醒按键，暂时未配置使用。
- 扩展板主要将核心板上的所有接口引下来，并提供一些必要的接口。
- USB从设备口J11：主要用来进行SAM-BA的编程。
- USB主设备接口J12：两路，可以支持U盘，usb网卡等多种设备
- 调试串口J1:用来输出调试打印信息，就是控制台
- 串口J2：该串口为全功能串口。

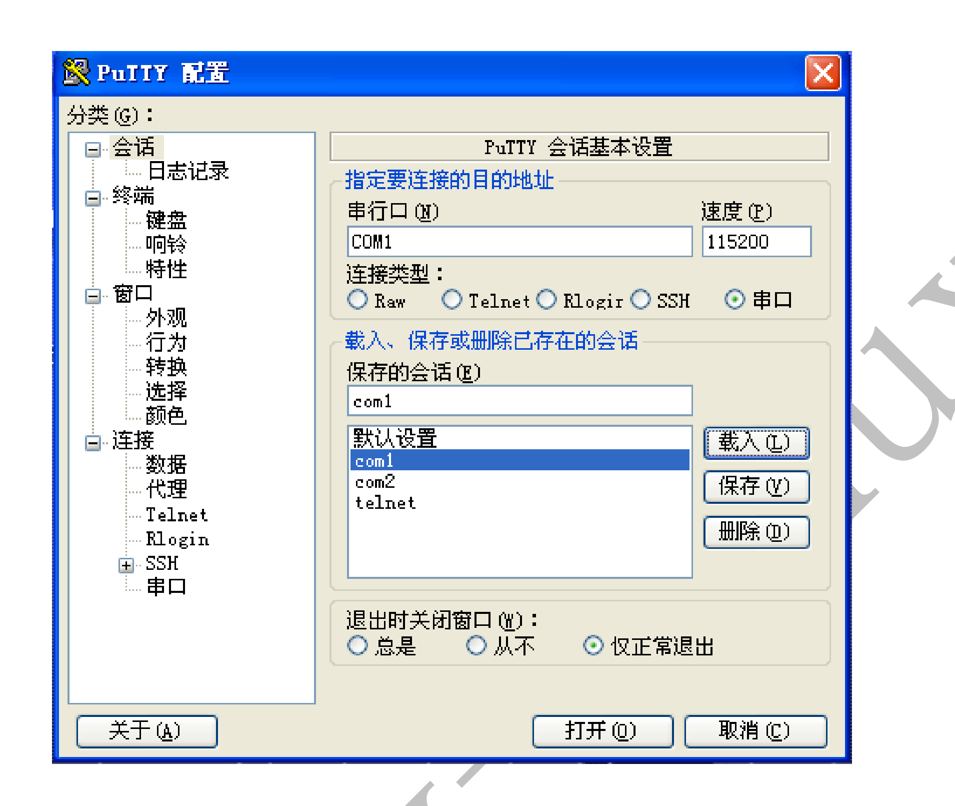
Tiny Linux开发板的特色，比别的开发板优秀的地方，相信我们，Tiny Linux的每一个程序都是Linux一线研发工程人员智慧的结晶，而不是简单的网络上流传的代码的收集整理移植者。

- Yaffs与JFFS根文件系统：
  我们提供两种这两种可写入的根文件系统，并提供一键烧写和一键恢复工具，产品的生产将非常方便，目前大部分开发板提供的是ROMFS\CRAMFS\RAMDISK,他们或者是不能对根文件系统进行写操作，或者是修改的配置重新上电会丢失。
- 在系统编程:
  提供工具实现在linux系统内对bootstrap、uboot、linux、根文件系统的在线升级，这对需要在线升级的项目来说是非常实用的。
- 超强的网络功能：
  支持有线与无线的网络设备，支持热插拔，支持dhcp,通过配置文件配置，能做到即插即用，免手动操作，做到桌面电脑一样智能。
- 网络工具完整：
  Tiny Linux开发板移植了完整的网络工具，并在研发环境中实践验证过，Tiny Linux支持Telnet、FTP、Gdbserver、BOA、wget。能实现网络下载等网络功能。
- 完整的linux研发流程
  我们从事嵌入式linux软件研发多年，积累了大量的研发、调试经验，我们提供基于linux的研发、调试、代码管理模式的整套解决方案，并提供文档支持，让您的软件开发犹如windows下普通的C语言开发一样方便。
- Java环境支持
  支持Java虚拟机，在目前的很多行业：例如数字电视，业内已经越来越认可Java作为研发语言，可以实现在嵌入式CPU上Java编程的理想。
- 超快的启动时间
  当我们使用YAFFS的文件系统时，我们可以保证从系统上电到进入登陆界面时间在（包括驱动加载等）控制在6秒左右，这在嵌入式linux界比较难得。

Tiny Board 使用手册

1 登陆系统

控制台串口通信配置为：波特率：115200，数据位：8，奇偶校验：无，停止位：1，数据流控：无，Tiny-Linux开发板默认登陆用户为:test，密码为123456。

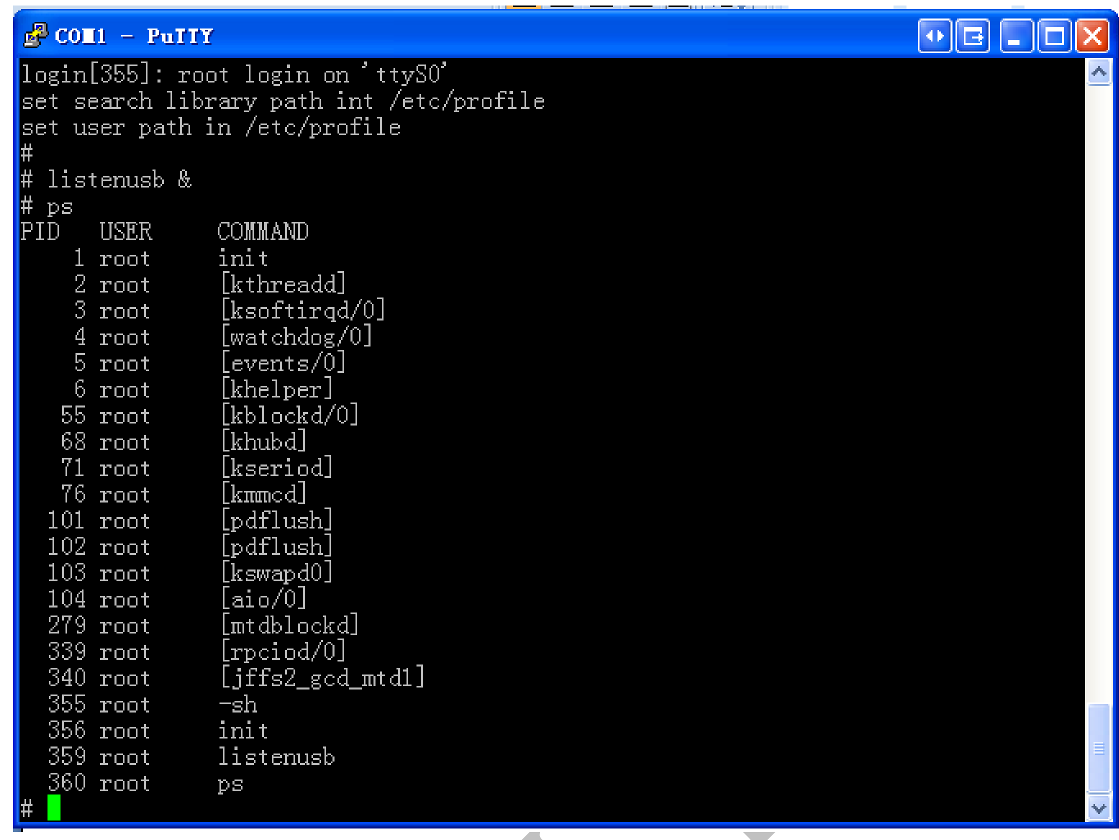



```
COM1 - PuTTY
/media/disk/tony/DM9601_2/dm/dm9601-2.6/dm9601.c: /media/disk/tony/DM9601_2/dm/dm9601-2.6/dm9601.c: v0
.0.6 2001/05/24 (C) 1999-2000 Petko Manolov (petkan@dce.bg)
usbcore: registered new interface driver dm9601
prism2usb_init: prism2_usb.o: 0.2.9 Loaded
prism2usb_init: dev_info is: prism2_usb
ident: nic h/w: id=0x8026 1.0.0
ident: pri f/w: id=0x15 1.1.2
ident: sta f/w: id=0x1f 1.8.1
MFI:SUP:role=0x00:id=0x01:var=0x01:b/t=1/1
CFI:SUP:role=0x00:id=0x02:var=0x02:b/t=1/1
PRI:SUP:role=0x00:id=0x03:var=0x01:b/t=1/4
STA:SUP:role=0x00:id=0x04:var=0x01:b/t=1/15
PRI-CFI:ACT:role=0x01:id=0x02:var=0x02:b/t=1/1
STA-CFI:ACT:role=0x01:id=0x02:var=0x02:b/t=1/1
STA-MFI:ACT:role=0x01:id=0x01:var=0x01:b/t=1/1
Prism2 card SN: 000000000000
usbcore: registered new interface driver prism2_usb

tiny_linux login: test
Password:
login[355]: root login on 'ttyS0'
set search library path int /etc/profile
set user path in /etc/profile
#
```

2 网络支持

开发板上电之后默认支持有线网卡与无线网卡， 默认网卡的即插即用没有打开，手动运行listenusb &将打开网卡的即插即用。如果采用即插即用的方式话，则所有的网络服务自动启动，



```
COM1 - PuTTY
login[355]: root login on 'ttyS0'
set search library path int /etc/profile
set user path in /etc/profile
#
# listenusb &
# ps
PID   USER     COMMAND
    1 root     init
    2 root     [kthreadd]
    3 root     [ksoftirqd/0]
    4 root     [watchdog/0]
    5 root     [events/0]
    6 root     [khelper]
   55 root     [kblockd/0]
   68 root     [khubd]
   71 root     [kseriod]
   76 root     [kmmcd]
  101 root     [pdflush]
  102 root     [pdflush]
  103 root     [kswapd0]
  104 root     [aio/0]
  279 root     [mtdblockd]
  339 root     [rpciod/0]
  340 root     [jffs2_gcd_mtd1]
  355 root     -sh
  356 root     init
  359 root     listenusb
  360 root     ps
#
```

2.1 无线网卡支持

此时，重新插拔一下网卡，控制台将打印出如下的信息：

```
add@/class/usb_endpoint/usbdev1.5_ep00
readlen =314
add@/devices/platform/at91_ohci/usb1/1-2/1-2:1.0
readlen =205
add@/class/net/wlan0
checked wlan0
prism2sta_ev_info: Unknown info type=0xfd42
ident: nic h/w: id=0x8026 1.0.0
ident: pri f/w: id=0x15 1.1.2
ident: sta f/w: id=0x1f 1.8.1
MFI:SUP:role=0x00:id=0x01:var=0x01:b/t=1/1
CFI:SUP:role=0x00:id=0x02:var=0x02:b/t=1/1
PRI:SUP:role=0x00:id=0x03:var=0x01:b/t=1/4
STA:SUP:role=0x00:id=0x04:var=0x01:b/t=1/15
PRI-CFI:ACT:role=0x01:id=0x02:var=0x02:b/t=1/1
STA-CFI:ACT:role=0x01:id=0x02:var=0x02:b/t=1/1
STA-MFI:ACT:role=0x01:id=0x01:var=0x01:b/t=1/1
Prism2 card SN: 000000000000
linkstatus=CONNECTED
udhcpc (v1.13.2) started
Sending discover...
Sending select for 192.168.1.102...
Lease of 192.168.1.102 obtained, lease time 86400
deleting routers
route: SIOCDELRT: No such process
adding dns 211.148.192.141
adding dns 210.21.196.6
readlen =242
add@/class/usb_endpoint/usbdev1.5_ep81
readlen =242
add@/class/usb_endpoint/usbdev1.5_ep02
readlen =242
add@/class/usb_endpoint/usbdev1.5_ep83

#
```

当打印出“check wlan0”时，表示检测到了无线网卡，后面的dhcp请求表示已经从DHCP路由器获得IP：192.168.1.102。DHCP的配置脚本为/etc/iw_get。根据自己的实际情况修改自己的接入点和密码,注意,如果无线路由器开通了MAC地址过滤的话,那你一定要将Tiny Linux的mac地址加入到路由器的信任地址内。

```
# cat /etc/iw_get
#!/bin/sh
#insmod p80211.ko
#insmod prism2_usb.ko
#ifconfig wlan0 down
iwconfig wlan0 essid "linksys"
iwconfig wlan0 key E2B6E73DAB
iwconfig wlan0 mode Managed
iwconfig wlan0 channel 6
#ifconfig wlan0 192.168.1.100

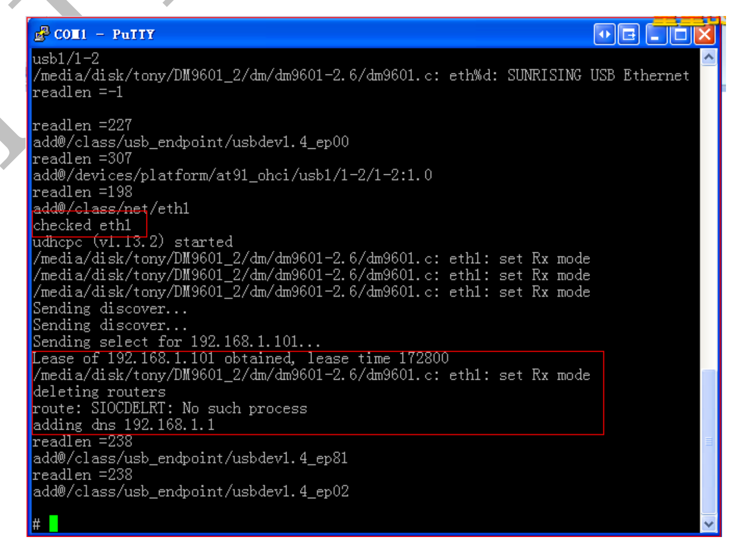
keys=`(ps -ef |grep "udhcpc" |grep -v "grep") | awk '{print $1}'`
#echo "kill udhcpc ${keys}"
for key in ${keys}
do
        #echo "kill ${key}"
        kill -9 $key
done

keys2=`(ps -ef |grep "vsftpd" |grep -v "grep") | awk '{print $1}'`
#echo "kill vsftpd ${keys2}"
for key2 in ${keys2}
do
        #echo "kill ${key2}"
        kill -9 $key2
done

keys3=`(ps -ef |grep "telnetd" |grep -v "grep") | awk '{print $1}'`
#echo "kill telnetd ${keys3}"
for key3 in ${keys3}
do
        #echo "kill ${key3}"
        kill -9 $key3
done
```

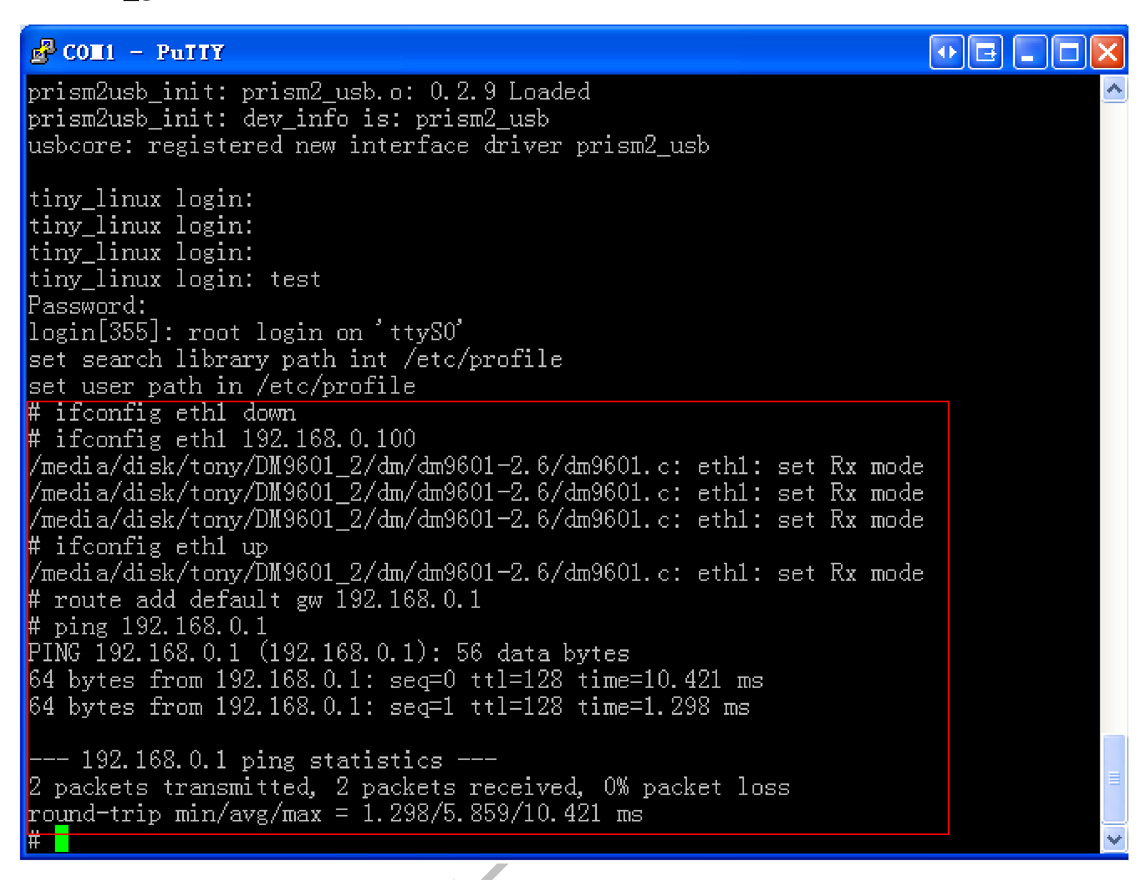
2.2 有线网卡支持

需先将USB网卡使用普通网线和路由器相连，然后将USB网卡插上Tiny Linux系统。注意，该方法只适用于与DHCP的路由器相连。



也可使用交叉网线与电脑直接相连，这样就不能使用dhcp，即需要手动来配置

网络地址。首先在windows上将系统的IP手动设置为192.168.0.1。然后在Tiny Linux上按照下面的逐步执行。但是telnetd、FTP、boa需要手动运行，运行格式请参照/etc/iw_get脚本。



3 网络工具使用

3.1 telnet服务

本开发板为了方便用户使用，移植成功telnetd服务器，已经可以使用telnet登陆终端。

telnet 192.168.0.100，也可用PUTTY工具，用户名与密码和控制台的一致。

```
192.168.0.100 - PuTTY

tiny_linux login: test
Password:
set search library path int /etc/profile
set user path in /etc/profile
# ls
#
#
#
#
#
#
#
# ls -al
```

3.2 ftp服务

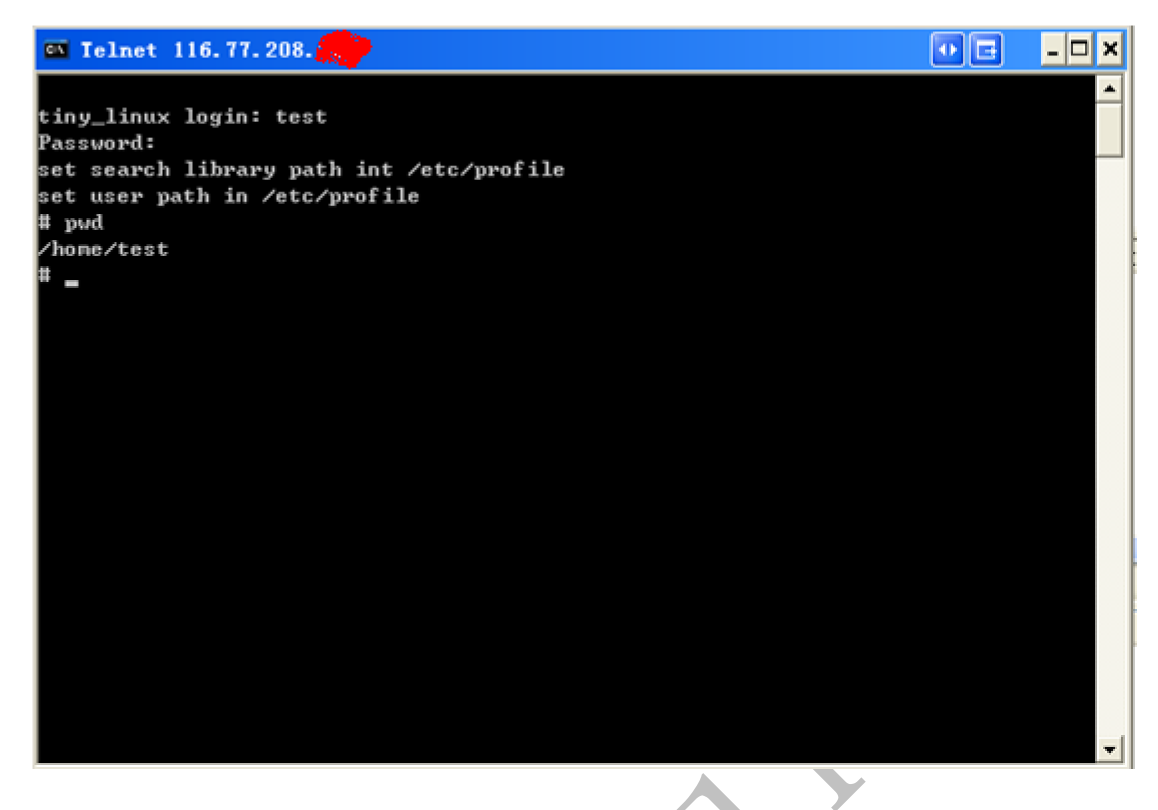
Tiny Linux上有完整的FTP服务器，可使用filezilla程序与服务器端交换数据。用户名与密码和控制台的一致。

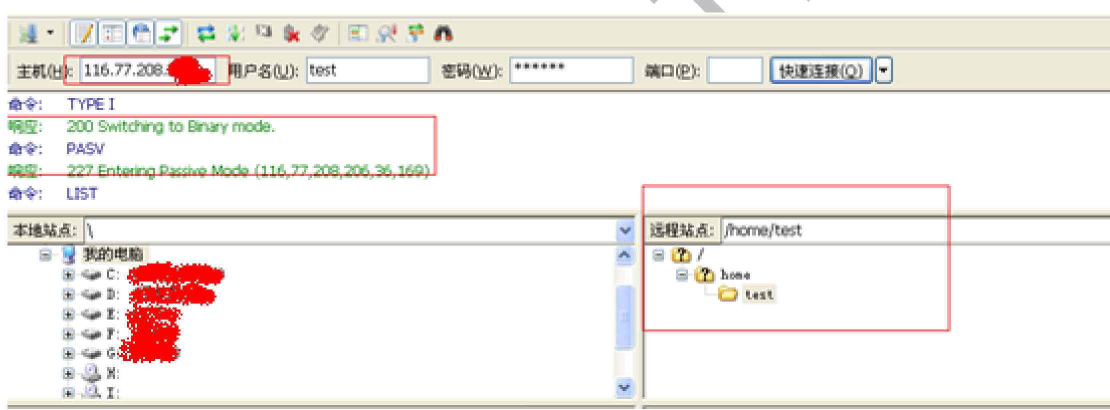


注意该FTP最大的亮点是支持Pasv模式，即外部网络可以通过路由器访问该FTP站点，首先要将路由器的下面选项打开。



这两者打开之后，我们就可以直接通过网络IP来远程访问我们的设备了。包括Telnet和FTP。



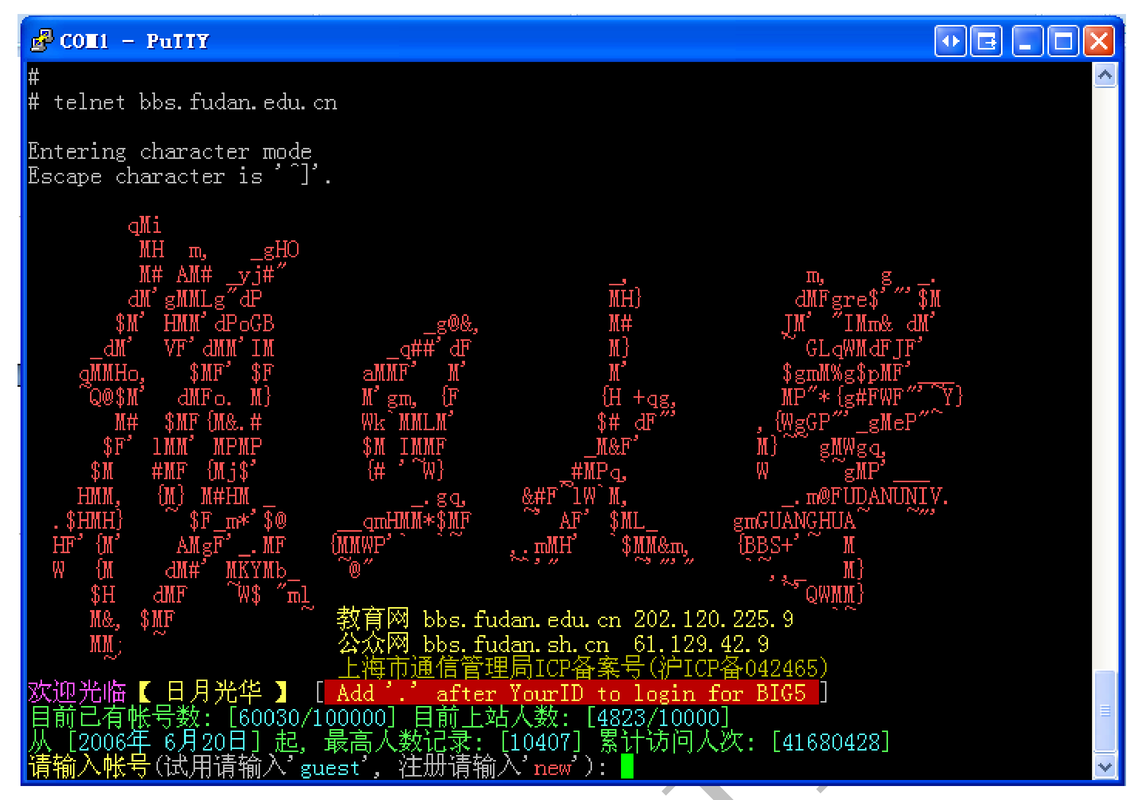


Tiny Linux支持域名解析:

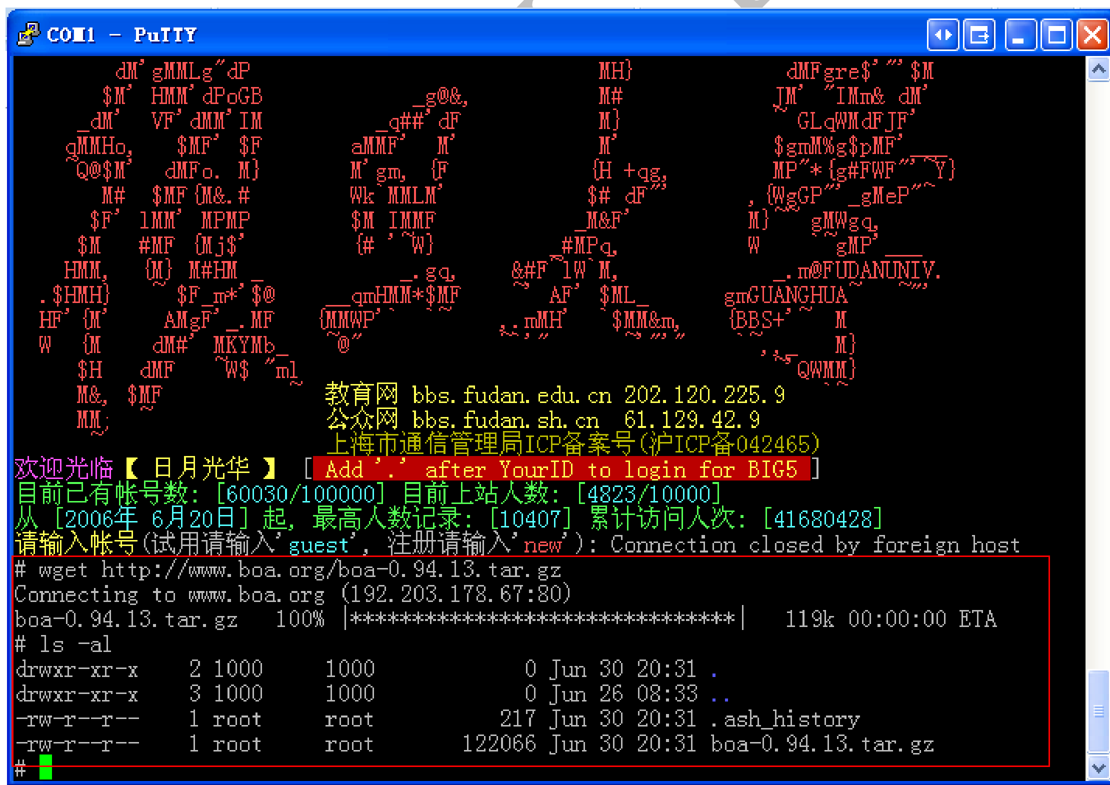
```
COM - PuTTY
# ping www.163.com
PING www.163.com (58.248.245.40): 56 data bytes
64 bytes from 58.248.245.40: seq=0 ttl=52 time=50.546 ms
64 bytes from 58.248.245.40: seq=1 ttl=52 time=44.312 ms
64 bytes from 58.248.245.40: seq=2 ttl=52 time=77.345 ms
64 bytes from 58.248.245.40: seq=3 ttl=52 time=63.353 ms

--- www.163.com ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 44.312/58.889/77.345 ms
```

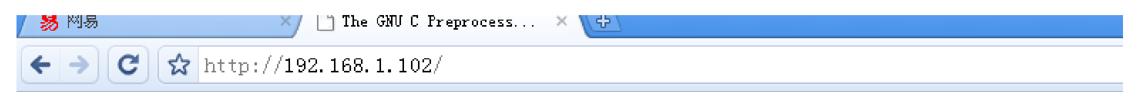
Tiny Linux支持远程登陆BBS:



Tiny Linux支持下载：



3.3 boa服务

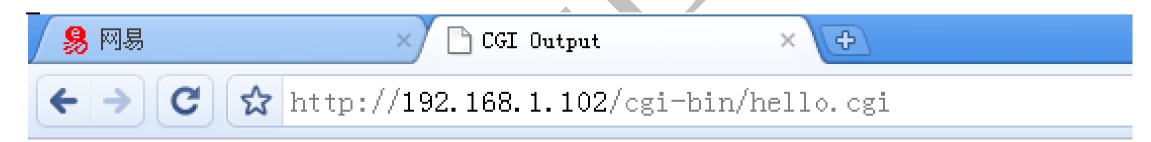


The GNU C Preprocessor Internals

Table of Contents

- The GNU C Preprocessor Internals
- 1 Cpplib—the GNU C Preprocessor
- Conventions
- The Lexer
  - Overview
  - Lexing a token
  - Lexing a line
- Hash Nodes
- Macro Expansion Algorithm
  - Internal representation of macros
  - Macro expansion overview
  - Scanning the replacement list for macros to expand
  - Looking for a function-like macro's opening parenthesis
  - Marking tokens ineligible for future expansion
- Token Spacing
- Line numbering
  - Just which line number anyway?
  - Representation of line numbers
- The Multiple-Include Optimization
- File Handling
- Concept Index

支持cgi编程:



Hello, world.

3.4 GDB远程调试

Gdbserver是一项非常伟大的发明，它让所有的交叉调试就如同单片机程序一样那么容易仿真。

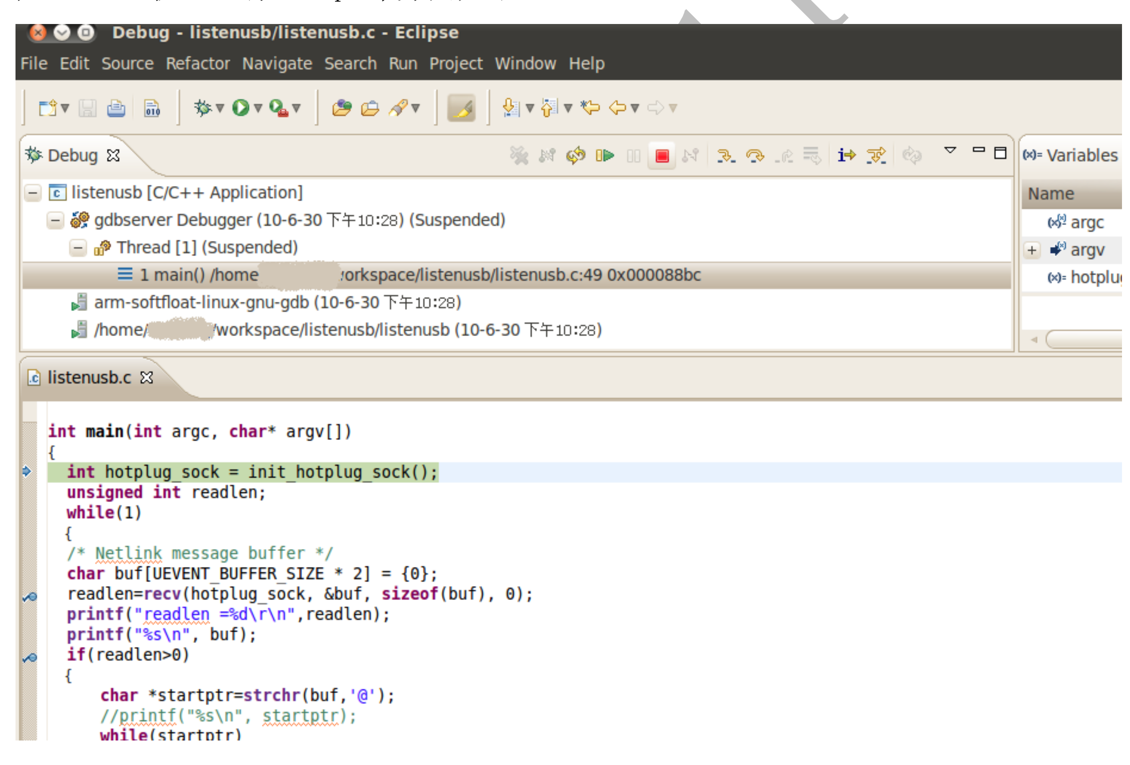
在Tiny Linux主机上运行，gdbserver 192.168.1.102:3345 listenusb
为远程调试的主机地址为192.168.1.102，listenusb为你需要调试的程序。

```
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 3.299/6.365/12.460 ms
# ifconfig
wlan0     Link encap:Ethernet  HWaddr 00:0F:66:B7:1D:87
          inet addr:192.168.1.102  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:289 errors:0 dropped:0 overruns:0 frame:0
          TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:83368 (81.4 KiB)  TX bytes:3990 (3.8 KiB)

#
#
#
# gdbserver 192.168.1.102:3345 listenusb
Cannot exec listenusb: Permission denied.
Process listenusb created; pid = 434

Child exited with retcode = 7f
No program to debug.  GDBserver exiting.
# chmod a+x listenusb
# chmod a+x listenusb
# gdbserver 192.168.1.102:3345 listenusb
Process listenusb created; pid = 440
Listening on port 3345
Remote debugging from host 192.168.1.100
```

在linux主机上运行eclipse,界面如下。



3.5 NFS服务。

在Tiny Linux上，支持NFS服务，在UBUNTU服务器上安装

apt-get install nfs-kernel-server

apt-get install nfs-common

在这里我们要建议您一句，建议普通的学习用户使用虚拟机的方式来建立linux研发环境，因为在软件研发方面linux非常方便，而在娱乐和其他方面，和windows

相比还是有很大的差距，对公司用户来说，建议使用服务器方式，这样一来编译器环境将会统一，二来方便研发管理。建议安装ubuntu。

修改服务器上的/etc/export文件，添加：

/home/test/kfboard　192.168.1.102(rw,sync,no_root_squash)

必须要加上最后一个配置，不然加载成功也将不能访问。

在开发板执行：

mount -t nfs -o nolock 192.168.1.103:/home/test/kfboard /mnt/flash0

则在开发板的/mnt/flash0看到NFS目录。

```
drwxr-xr-x    8 1000     1000         4096 Nov 20  2008 phoneme_advanced_mr2
drwxr-xr-x   14 1000     1000         4096 Jun 26 08:30 root_fs
-rwx------    1 1000     1000      2744320 Sep  5  2009 root_fs.img
-rw-r--r--    1 1000     1000      2756608 Jun 27 03:21 root_fs_cramfs.img
-rw-r--r--    1 1000     1000      3538944 Jun 29 14:01 root_fs_jffs.img
-rw-------    1 1000     1000      7406784 Jun 29 13:54 root_fs_yaffs.img
# cat /proc/mounts
rootfs / rootfs rw 0 0
/dev/root / jffs2 rw 0 0
proc /proc proc rw 0 0
devpts /dev/pts devpts rw 0 0
tmpfs /dev/shm tmpfs rw 0 0
sysfs /sys sysfs rw 0 0
/dev/mtdblock2 /mnt/flash2 yaffs rw 0 0
192.168.1.103:/home/[illegible]/kfboard /mnt/flash0 nfs rw,vers=3,rsize=32768,wsize=32768,hard,nointr,noloc
k,proto=udp,timeo=7,retrans=3,sec=sys,addr=192.168.1.103 0 0
#
```

4 关于一键恢复和文件系统的烧写

注意：没有百分之百的把握，请勿轻易改动我们提供的bootstrap，因为这样的话，很有可能导致系统不能正常启动，而Tiny Linux开发板又不能进入SAM-BA模式，我们提供的Bootstrap在系统上电启动的时候同时按住S1和S2键，将清除Bootstrap，再次启动系统将进入SAM-BA模式，如果您不小心修改了bootstrap而导致不能进入SAM-BA模式，而系统又无法正常启动，这时您需要将NAND FLASH的cs引脚上的电阻断开，即R22，以使Tiny Linux在上电的时候能够进入SAM-BA模式，然后接上R22，以使其能都对NANDFLASH进行操作，如果没有硬件条件，将需要返回给我们来帮您处理，所以，没有十足的把握，请不要轻易的修改我们提供的BootStrap代码。

4.1 一键恢复系统

Tiny Linux开发板提供了两套文件系统，JFFS与YAFFS，这两套根文件系统都是可读可写的，两者各有优点，如果你开发的系统需要处理的数据量比较大，这样你可能需要使用YAFFS文件系统，如果你需要的是一个非常稳定的系统，同时您的系统又可能经常存在掉电的情况的话，请使用JFFS系统。

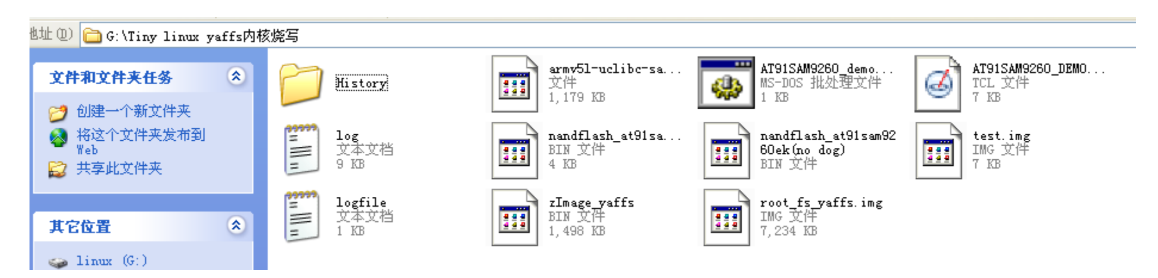
SAM-BA的Windows安装版指引：首先执行安装程序：Install AT91-ISP v1.12.exe，然后将我们提供的SAM-BA v2.8拷贝覆盖C:\Program Files\ATMEL Corporation\AT91-ISP v1.12\SAM-BA v2.8这个目录即可。

:\工具\SAM-BA\WINDOWS

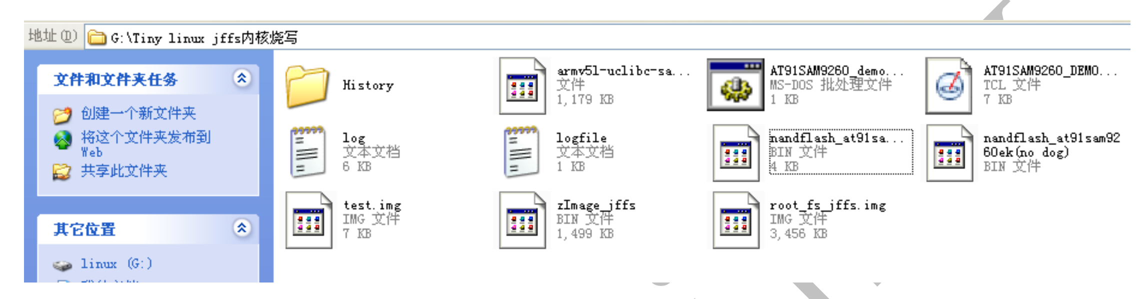
Install AT91-ISP v1.12

SAM-BA v2.8
WinRAR 压缩文件
4,906 KB

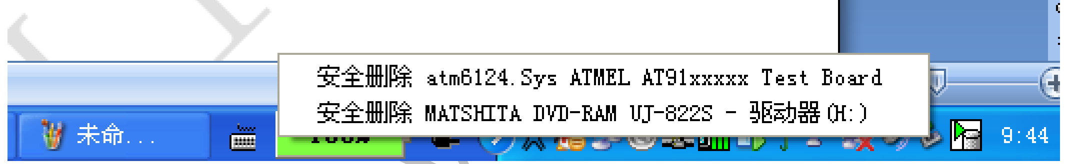
YAFFS一键烧写文件系统：



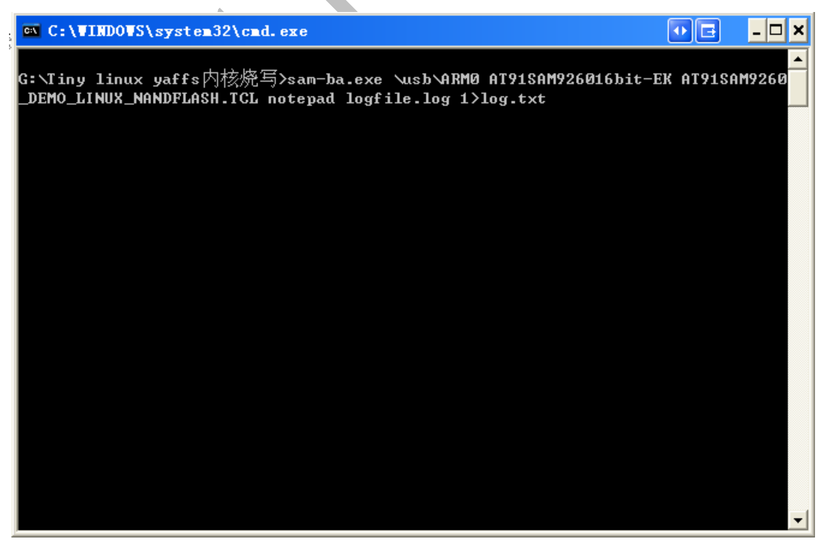
JFFS一键烧写文件系统：



首先Tiny Linux进入SAM-BA模式，用USB设备连接线将主机和J11相连，PC端应该出现检测到atmel设备的字样，如下图，如果没有，说明您的设备没有进入SAM-BA模式，这时您需要重启开发板并按住S1和S2键，3秒后再次断电重启，Tiny Linux将进入SAM-BA模式。



进入SAM-BA模式之后，只需要鼠标点击YAFFS或者JFFS文件夹里面的批处理命令即可，等待该命令程序执行完毕，命令执行过程中请勿关闭该窗口，所有的打印信息都输出到同文件夹下的log文件中，如果需要可以仔细察看。



这种方式就可以直接实现产品的批量生产。并可对生产制造是否成功进行检测，不同于其他开发板使用UBOOT烧写系统的方式，那种方式不可能适合于批量生产，生

产效率低。

Log文件的内容：

```
log - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
-I- Waiting ...
connection : \usb\ARM0
board : AT91SAM926016bit-EK
target(handle) : 18022016
read chip ID : 0x00000010 at addr: 0xFFFFEE40
read chip ID : 0x019803A2 at addr: 0xFFFFF240
-I- Found processor : AT91SAM9260 (0x019803A0)
-I- Loading applet isp-extram-at91sam9260.bin at address 0x200000
-I- init result =0
-I- Memory Size : 0x2000000 bytes
-I- Buffer address : 0x2007C4
-I- Buffer size: 0x0 bytes
-I- Applet initialization done
-I- External RAM initialized
script file : AT91SAM9260_DEMO_LINUX_NANDFLASH.TCL
u-boot file: logfile.log
diskSize: 0x007104C0
-I- === Performs sanity checks===
-I- === Initialize the NAND access ===
-I- NANDFLASH::Init (trace level : 3)
-I- Loading applet isp-nandflash-at91sam9260.bin at address 0x20000000
-I- init result =0
-I- Memory Size : 0x8000000 bytes
-I- Buffer address : 0x20004DFC
-I- Buffer size: 0x20000 bytes
-I- Applet initialization done
-I- === Erase all the NAND flash blocs and test the erasing ===
-I- GENERIC::EraseAll erase flag=0
-I- === Load the bootstrap: nandflash_at91sam9260ek in the first sector ===
-I- Send File nandflash_at91sam9260ek.bin at address 0x00000000
GENERIC::SendFile nandflash_at91sam9260ek.bin at address 0x0
-I- File size : 0xF50 byte(s)
-I-     Writing: 0xF50 bytes at 0x0 (buffer addr : 0x20004DFC)
```

4.2 在系统编程

Tiny linux的MTD分区安排如下：

Mtdblock0 bootstrap

Mtdblock1 kernel

Mtdblock2 root file system

我们提供工具，能够对MTD设备进行在系统的编程，即对bootstrap、内核、文件系统都能进行在线升级，我们在linux系统里通过网络接受最新的内核、文件系统等，保存在系统中，然后通过我们提供的工具，在linux环境下对MTD设备进行更新的操作，重启后系统工作正常。

升级YAFFS分区：先使用NFS将升级的文件加载进来，然后擦除FLASH，再将需要的文件写入。注意，写入yaffs文件分区时必须使用writeyaffs方法。

```
# umount /mnt/flash0
# mount -t nfs -o nolock 192.168.1.104:/home/t[illegible]g/kfboard /mnt/flash0
# dataflash_ioctl erase /dev/mtd2
info.size=10485760
 info.erasesize=131072
 info.writesize=2048
 info.oobsize=64
regcount=0
Performing Flash Erase of length 131072 at offset 0x9e0000 done
erase!
# dataflash_ioctl writeyaffs /dev/mtd2 /mnt/flash0/root_fs_yaffs.img
info.size=10485760
 info.erasesize=131072
 info.writesize=2048
 info.oobsize=64
regcount=0
write yaffs!
The input file image len is 7406784!
Write buf is ok!
write mtd device ok!startaddr=7404672
#
```

升级JFFS2分区：先使用NFS将升级的文件加载进来，然后擦除FLASH，再将需要的文件写入。注意，写入JFFS2文件分区时必须使用write方法。

```
# dataflash_ioctl erase /dev/mtd2
info.size=10485760
 info.erasesize=131072
 info.writesize=2048
 info.oobsize=64
regcount=0
Performing Flash Erase of length 131072 at offset 0x9e0000 done
erase!
# dataflash_ioctl write /dev/mtd2 /mnt/flJFFS2 notice: (351) jffs2_get_inode_nod
es: Node header CRC failed at 0x33b6ac. {ffff,ffff,ffffffff,ffffffff}
JFFS2 warning: (351) jffs2_do_read_inode_internal: no data nodes found for ino #
330
JFFS2 notice: (351) jffs2_get_inode_nodes: Node header CRC failed at 0x33b794. {
ffff,ffff,ffffffff,ffffffff}
JFFS2 warning: (351) jffs2_do_read_inode_internal: no data nodes found for ino #
332
# dataflash_ioctl write /dev/mtd2 /mnt/flash0/root_fs_jffs.img
info.size=10485760
 info.erasesize=131072
 info.writesize=2048
 info.oobsize=64
regcount=0
write!
The input file image len is 3538944!
Write buf is ok!
write mtd device ok!startaddr=3536896
#
```

同时YAFFS和JFFS两者之间还可以互相升级。

如何让系统重启后能进入SAM-BA模式：

```
linkstatus=CONNECTED
# dataflash_ioctl erase /dev/mtd0
info.size=131072
 info.erasesize=131072
 info.writesize=2048
 info.oobsize=64
regcount=0
Performing Flash Erase of length 131072 at offset 0x0 done
erase!
# reboot
```

由于我们的分区表已经将BOOTSTAP、kernel、rootfs都分区，所以可以自由实现三者的程序升级。

5 JAVA虚拟环境

当今JVM种类很多，如IBM JVM、MS JVM等等。但要把它用到ARM上，并非易事。几经过测试和对比，最终还是选择了SUN的JVM，其中，J2ME（Java 2 Platform MicroEdition）是专为移动设备设计的，在移动通信设备上已经有了很广的应用，非常适合我们在嵌入式中使用。J2ME根据硬件资源的限制，分为CLDC（Connected Limited DeviceConfiguration）和CDC（Connected Device Configuration），无论是CLDC还是CDC，其核心都是Java虚拟机。

Atmel 的9260支持java指令，有公司使用此做机顶盒的设计方案，所以我们在Tiny Linux上移植了一个JAVA虚拟机。JAVA虚拟机存放于/home/j2me，CVM为JAVA虚拟机。我们运行测试程序，验证虚拟机的环境是否正常。

```
# jvm -version
Product: phoneME Advanced (phoneme_advanced_mr2-b97)
Profile: CDC Specification 1.1
JVM:     CVM phoneme_advanced_mr2-b97 (mixed mode)
# jvm -cp testclasses.zip HelloWorld
Hello world.
#
```

在简单的测试之后，要进行全面的虚拟机测试:

Jvm -cp testclasses.zip Test

```
c4 = class [[LC;
c1.modifiers = 1041
c2.modifiers = 1041
c3.modifiers = 1040
c4.modifiers = 1040
*FloatMIN =1.4E-45
*FloatMAX =3.4028235E38
FloatMIN (the int bits) =1
FloatMAX (the int bits) =2139095039
java.lang.IllegalArgumentException: too many dimensions
        at java.lang.reflect.Array.multiNewArray(Native Method)
        at java.lang.reflect.Array.newInstance(Unknown Source)
        at Test.testDeepArrayConstruction(Unknown Source)
        at Test.main(Unknown Source)
        at sun.misc.CVM.runMain(Unknown Source)
Constructed an object of type [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LTest;
finally!
Requesting GC with a latency request of 2 seconds
(Turning GC tracing on)
Sleeping 5 seconds, and waiting for GC's
Woke up! Cancelling latency request
.............
*CONGRATULATIONS: test Test completed with 411 tests passed and 0 failures
*Output lines starting with a * should be checked for correctness
*They can be compared to src/share/javavm/test/TestExpectedResult
#
```

测试显示所有JAVA环境测试项都通过。

我们以LINUX环境下的JAVA程序开发为例，说明JAVA编译的注意点，必须要使用1.4版本，否则将在Tiny Linux上运行报错。

```
es
u:~/kfboard$ javac -source 1.4 hello.java
u:~/kfboard$ javac -source 1.4 hello.java
u:~/kfboard$
```

在开发板上如下运行：

```
tiny_linux login: test
Password:
6Login incorrect
tiny_linux login: test
Password:
login[352]: root login on 'ttyS0'
set search library path int /etc/profile
set user path in /etc/profile
# cd ../j2me/
# jvm hello
Welcome to tiny Linux world
#
```





天鹰科技出品

嵌入式Linux产品研发、方案设计提供商

期待与您的合作